# Fișa 01 C# – introducere

**Prezentare mediu + instrucțiuni de intrare / ieșire**

a) Citire date de la tastatură:

//șmecherie pentru afișare diacritice în consolă

//(setăm setul de caractere al consolei la UTF8)

Console.OutputEncoding = System.Text.UTF8Encoding.UTF8;

//definire variabile

int i;

float f;

string s;

//citire date

Console.Write("\nIntroduceți un număr întreg : ");

i = int.Parse( Console.ReadLine() );

//șirul citit se transformă în nr. întreg cu funcția int.Parse()

Console.Write("\nIntroduceți un număr real : ");

f = float.Parse(Console.ReadLine());

//șirul citit se transformă în nr. real cu funcția float.Parse()

Console.Write("\nIntroduceți un șir : ");

s = Console.ReadLine();

//gata

Console.Write("\n\n\nGata! ");

b) Afișare date pe ecran:

//afișare

Console.WriteLine("\nAți introdus numărul întreg {0}", i);

Console.WriteLine("\nAți introdus numărul real {0}", f);

Console.WriteLine("\nAți introdus șirul \"{0}\"", s);

În instrucțiunile de mai sus notația {0} semnifică „poziția la care se va afișa prima valoare” din cele care urmează după virgulă. Notația \" se folosește pentru a afișa ghilimele.

**Tratare excepții**

Ce se întâmplă dacă în programul de mai sus utilizatorul introduce un text când i se cere un număr întreg?

O ***excepție*** este un eveniment care se produce din motive independente de voința programatorului: utilizatorul face prostii sau acțiuni neprevăzute, defecțiuni hardware etc. Limbajul C# are un mecanism de tratare a excepțiilor care ne permite să includem în program un grup de instrucțiuni care se execută în cazul în care se produce o excepție. Folosind acest mecanism programul de mai sus se poate scrie în forma:

//șmecherie pentru afișare diacritice în consolă

//(setăm setul de caractere al consolei la UTF8)

Console.OutputEncoding = System.Text.UTF8Encoding.UTF8;

//definire variabile

int i;

float f;

string s;

try

{

//citire date

Console.Write("\nIntroduceți un număr întreg : ");

i = int.Parse(Console.ReadLine());

//șirul citit se transformă în nr. întreg cu funcția int.Parse()

Console.Write("\nIntroduceți un număr real : ");

f = float.Parse(Console.ReadLine());

//șirul citit se transformă în nr. real cu funcția float.Parse()

Console.Write("\nIntroduceți un șir : ");

s = Console.ReadLine();

//afișare

Console.WriteLine("\nAți introdus numărul întreg {0}", i);

Console.WriteLine("\nAți introdus numărul real {0}", f);

Console.WriteLine("\nAți introdus șirul \"{0}\"", s);

}

catch (Exception ex)

{

Console.WriteLine("Hopa! Ai făcut o boacănă : " + ex.Message);

}

//gata

Console.Write("\n\n\nGata! ");

În acest caz se încearcă execuția comenzilor de citire și afișare (blocul de instrucțiuni try{…}). Dacă nu se produc excepții programul se termină cu bine. Dacă ceva nu este în regulă se va produce o excepție (se zice că programul „aruncă” o excepție) care va fi „prinsă” de blocul catch{…} și execuția va sări la instrucțiunile din acesta după care programul se va termina. Blocul catch{…} primește de la sistemul de execuție și câteva date despre excepția produsă, care se memorează într-un obiect de tip Exception și pot fi folosite pentru a afișa informații despre excepție (în exemplu mesajul de eroare).

**Probleme:**

1. Concepeți un program care afișează pe ecran suma a două numere reale citite de la tastatură.
2. Concepeți un program care afișează pe ecran maximul a două numere reale citite de la tastatură.
3. Concepeți un program care afișează pe ecran c.m.m.d.c. și c.m.m.m.c. a două numere întregi citite de la tastatură.
4. Concepeți un program care citește un număr natural de la tastatură și verifică dacă este prim. *Pentru punct în plus rezolvați programul folosind o funcție „Prim”. Indicație: studiați unde și cum e definită funcția Main(). Care este semnificația cuvântului cheie* static*?*